AMENDMENTS TO THE CLAIMS

1-22   (Cancelled)

23.   (Previously Presented)   A system comprising:

a memory to store 32-bit architecture instructions;

an expanded logical register set including more than eight logical registers of a first type;

a bus coupled to said memory; and

32-bit architecture instruction set decoding logic coupled to the expanded logical register

set and the memory via the bus,

wherein the 32-bit architecture instruction set decoding logic is to determine whether a

mod field of a ModR/M byte of a 32-bit architecture instruction contains a predetermined value,

to determine whether an r/m field of the ModR/M byte of the 32-bit architecture instruction

contains a predetermined value, to determine whether an index field of a scale index base (SIB)

byte contains a predetermined value, and to decode an expanded logical register identifier.

24.   (Previously Presented) The system of claim 23, wherein said expanded logical register

set includes up to sixteen logical registers of a first type.

25.   (Previously Presented) The system of claim 23, wherein said expanded logical register

set includes up to thirty-two logical registers of a first type.

26.   (Previously Presented) The system of claim 23, wherein the more than eight logical

registers of a first type include more than eight logical general integer registers.

27.   (Previously Presented) The system of claim 23, wherein the more than eight logical

registers of a first type include more than eight logical multimedia extension technology

registers.

28.   (Currently Amended) The system of claim 23, wherein the more than eight logical

registers of a first type include more than eight logical ~~SIMD (single-instruction-multiple-data)~~

<u>single instruction multiple data (SIMD)</u> floating-point registers.

29.   (Previously Presented) The system of claim 23, further comprising:

expanded register set decoding logic, coupled to said 32-bit architecture instruction set

decoding logic, to determine whether an instruction includes an at least four-bit register

identifier, the at least four-bit register identifier to specify one logical register of said expanded

logical register set.

30.   (Previously Presented) The system of claim 29, wherein said expanded register set

decoding logic is to decode an at least four-bit identifier.

31.   (Previously Presented) The system of claim 30, wherein said expanded register set

decoding logic is to decode the at least four-bit register based at least in part on at least four bits

of a scale index base (SIB) byte of the instruction.

32.   (Previously Presented) The system of claim 31, wherein said expanded register set

decoding logic is to decode the at least four-bit register based at least in part on at least one bit of

a scale field of the scale index base (SIB) byte of the instruction.

33.   (Previously Presented) The system of claim 29, wherein said expanded register set

decoding logic is to decode an at least five-bit register identifier.

34.   (Previously Presented) The system of claim 33, said expanded register set decoding logic

is to decode the at least five-bit register based at least in part on five bits of a scale index base

(SIB) byte of the instruction.

35.   (Previously Presented) The system of claim 34, wherein said expanded register set

decoding logic is to decode the at least five-bit register based at least in part on two bits of a

scale field of the scale index base (SIB) byte of the instruction.

36.   (Currently Amended)   A set of instructions residing in a computer readable storage

medium, said set of instructions capable of being executed by a storage controller to implement a

method for processing data, the method comprising:

        determining ~~whether~~ that a mod field of a ModR/M byte of a 32-bit architecture

instruction contains a value selected from the values of 01B, 10B and 00B;

        determining ~~whether~~ that an r/m field of the ModR/M byte of the 32-bit architecture

instruction contains a value of 100B;

        determining ~~whether~~ that an index field of a scale index base (SIB) byte contains a value

of 100B;

decoding an at least four-bit logical register identifier in response to said values being

contained in said mod field, said r/m field, and said index field; and

retrieving a value from a register corresponding to said at least four-bit logical register

identifier.

37.   (Previously Presented) The set of instructions of claim 36, wherein decoding the at least

four-bit logical register identifier is based at least in part on at least one bit of a scale field of a

scale index base (SIB) byte of the 32-bit architecture instruction.

38.   (Previously Presented) The set of instructions of claim 36, wherein the at least four-bit

logical register identifier is an at least five-bit logical register identifier.

39.   (Previously Presented) The set of instructions of claim 38, wherein decoding an at least

five-bit logical register identifier is based at least in part on two bits of a scale field of a scale

index base (SIB) byte of the 32-bit architecture instruction.

40.   (Previously Presented)   A processor comprising:

a logical register set including more than eight logical registers of a first type;

instruction decoding logic coupled to the logical register set, said instruction decoding

logic including:

a first comparator to determine whether a mod field of a ModR/M byte of a 32-

bit architecture instruction contains a value selected from the values of 01B, 10B, and

00B;

a second comparator to determine whether an r/m field of the ModR/M byte of

the 32-bit architecture instruction contains a value of 100B;

a third comparator to determine whether an index field of a scale index base (SIB)

byte contains a value of 100B; and

an instruction decoder to identify a register of said logical register set based on

said values of said mod field, said r/m field and said index field.

41.   (Currently Amended) The processor of claim 40, wherein ~~the~~ said instruction decoder ~~to~~

~~decode an at least four-bit logical register identifier~~ is to decode ~~the~~ an at least four-bit logical

register identifier based at least in part on at least one bit of a scale field of a scale index base

(SIB) byte of the 32-bit architecture instruction.